Build Pages for MyEShop with HTML and CSS

7

LEARNING OBJECTIVES

After completing this chapter, you will be able to understand:
- Setup and integration of elements to create HTML pages.
- Entity relationship diagram.
- Development of web application using HTML and CSS.
- Use of any text editor to create HTML pages.
- How to add CSS on HTML elements.

7.1 | Setting up Environment

For HTML and CSS you do not really need to setup any environment. You can use any text editor you like. You can use the same Eclipse editor you have used in Chapter 3.

7.2 | Identify the Pages

Let us explore our entity relationship diagram again. See Figure 7.1 which we have presented in Chapter 2 (Figure 2.1) when we worked on architectural designs.

As per the entity relationship diagram (ERD), we have identified at least the following main entities:

Customer	Payment	Inventory
Address	Product	Catalogue
Shipping	Currency	Warehouse
Invoice	Shopping Cart	Vendor
Offer		

We have used these entities to create a page flow diagram in Chapter 2 (Figure 2.8); it is reproduced in Figure 7.2 which shows order flow for ease of understanding.

Figure 7.2 gives us an idea about which pages we should create for one particular use case. There are multiple use cases you can think of in the system. According to this use case, the flow starts from a main webpage, that is, the home page of *MyEShop* site. This page has astore-like look and feel. It also allows users to go to the login page or register page in order to access their personal dashboard. Now let us first start building the home page.

7.3 | Getting Started with HTML Pages

In this section, we will create a page which the user will visit as soon as he/she hits the website URL in a browser window. For this we will use a text editor and save this file with the name "*home.html*".
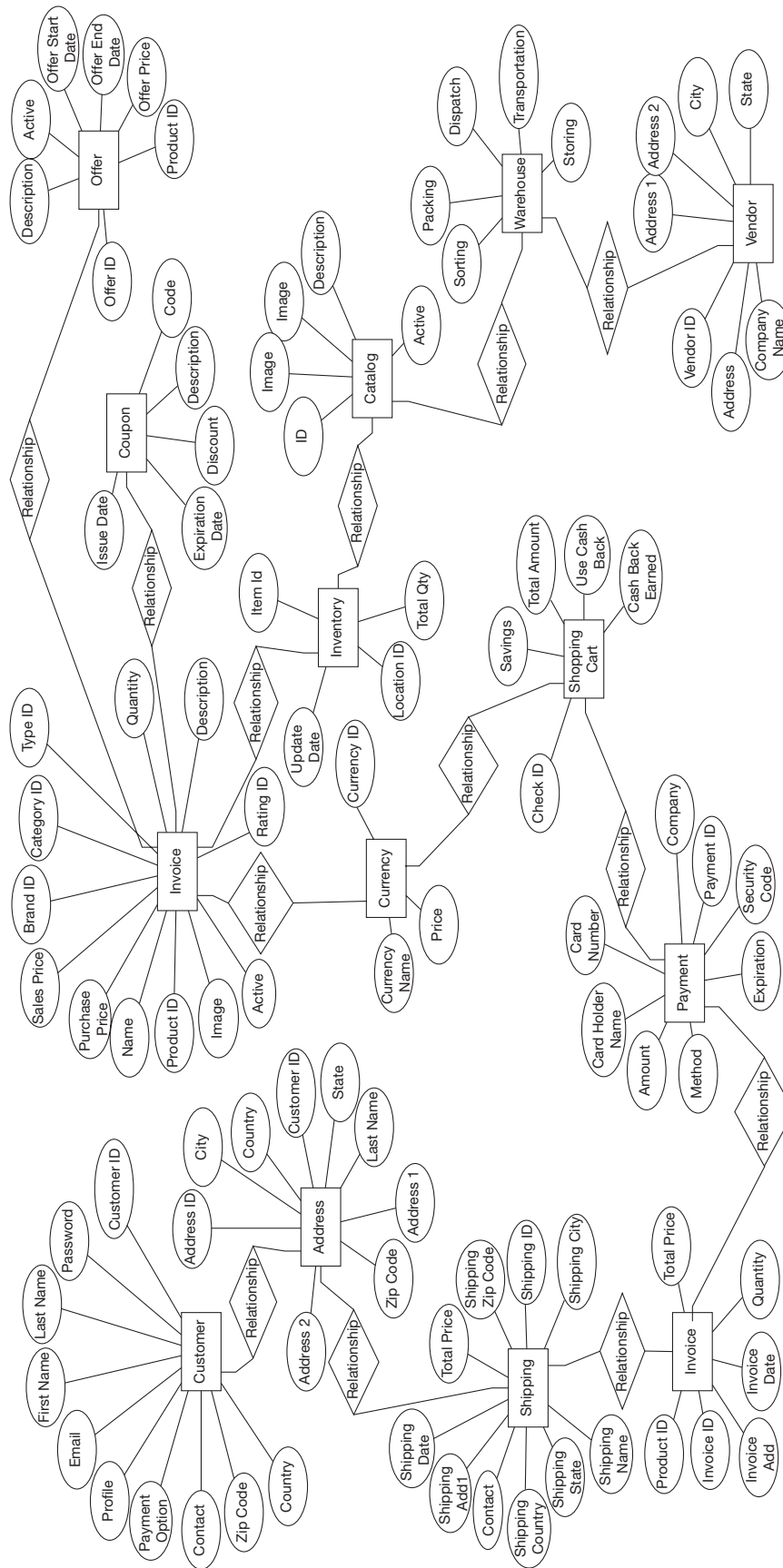
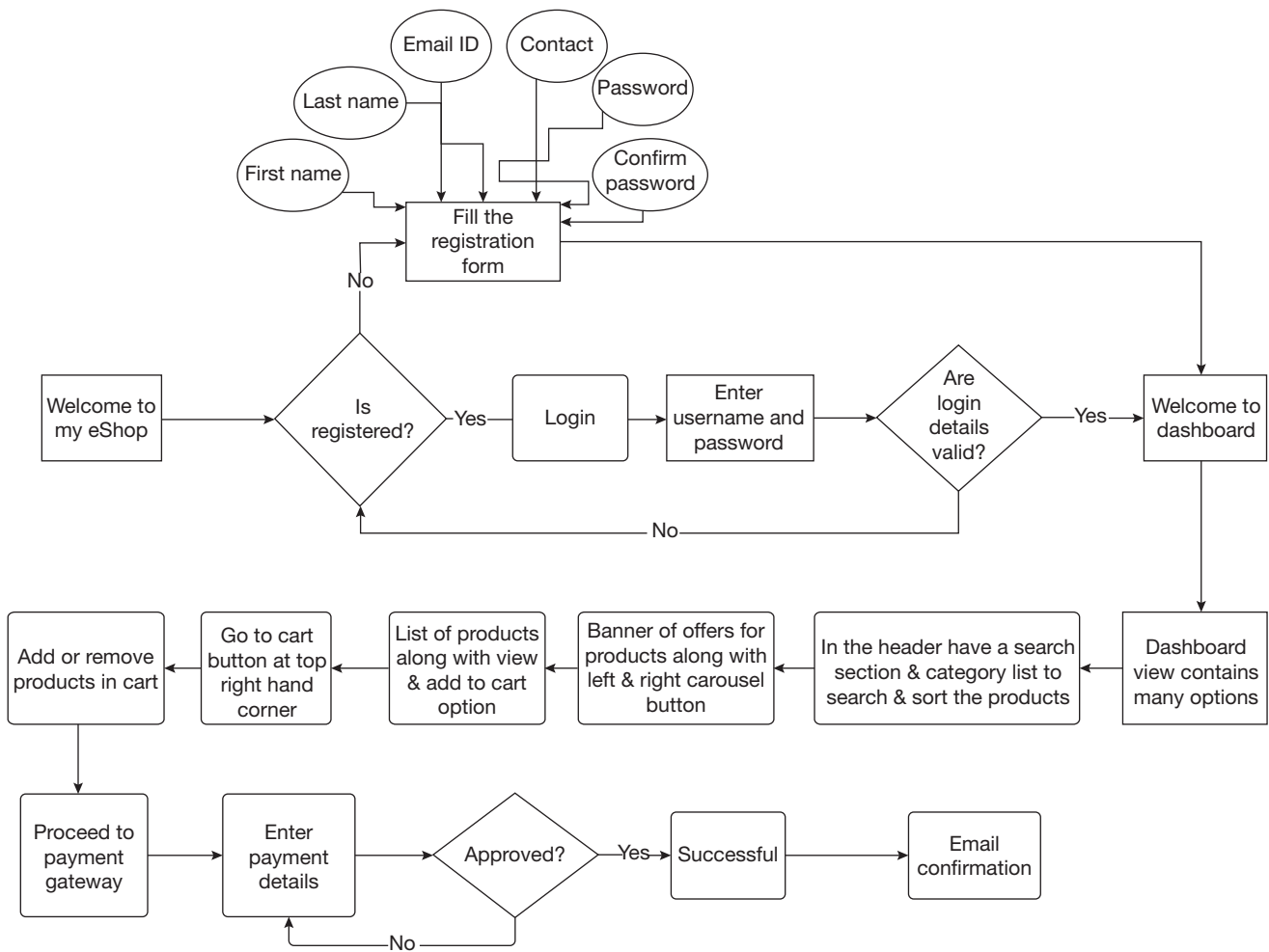**Figure 7.1** Entity relationship diagram of e-commerce entities.

Figure 7.2 Login, registration, search, purchase, and payment flowchart.

7.3.1 Home Page

After saving the file, let us continue working on the page "*home.html*".We now need to add various HTML elements to make it functional.

First, we will start with standard HTML structure that we have learned in Chapter 3.

```
<!DOCTYPEhtml>
<html>
    <head>
        <title>My First HTML Example</title>
    </head>
    <body>
        <h1>Lets Learn HTML History</h1>
        <p>HTML is the most popular language for webpages that was created in early
1990s</p>
    </body>
</html>
```

Now let us replace the text with our required inputs.

```
<!DOCTYPEhtml>
<html>
    <head>
        <title>MyEShop</title>
    </head>
    <body>
        <h1>Welcome to MyEShop</h1>
        <p>Buy Any Product you Like at a Reasonable Price</p>
    </body>
</html>
```

If you find difficulty understanding the above code, you can refer Chapter 3 where we have explained this in detail.

Now let us expand this further by adding one element type at a time.

7.3.2 Header

In this section we will create a header of the website that contains various elements such as logo, cart, navigation menu bar,and shopping cart.

7.3.3 Table

Let us add a table structure using the following code:

```
<table><tr><td></td></tr></table>
```

The explanation of the above elements is as follows:

<table>	HTML table element
<tr>	Table row
<td>	Table data/cell
</table>	Table closing tag
</tr>	Table row closing tag
</td>	Table cell closing tag
<th>	Table header tag
</th>	Table header closing tag

7.3.4 Logo

Let us add a logo image. We can use any geometric shape, preferably a circle, a square, or a rectangle.

```
<img src="myeshoplogo.png" alt="myeshop logo" width="500px"height= "600px">
```

<img>	Images are defined; contains only attributes and does not have a closing tag.
Src	Specifies the URL of the image.
Alt	Provides an alternate text for an image. If image does not get rendered, the text in alt tag is shown instead.
Height	Defines height of the image in pixels.
Width	Defines width of the image in pixels.

7.3.5 Search Bar

Let us add an input tab for search bar; we use a rectangular geometric shape.

```
<form><input type="text" placeholder="Search.."></form>
```

<form>	Defines a form that is used to collect user input.
<input>	Takes input type element.
<input type= "text"	Allows text input.
Placeholder	Gives provision to show a short descriptive text.
</form>	Form closing tag.

7.3.6 Category List

Create a tab dedicated for listing categories of products. It should allow single selection. However, if you like to add multiple selection option, you just need to add "multiple" word to <select> tag like <select multiple>.

```
<select>
    <option>All</option>
    <option>CLOTHES </option>
    <option>FOOD AND BEVERAGES </option>
    <option>HEALTH & BEAUTY </option>
    <option>SPORTS & LEISURE </option>
    <option>BOOKS & ENTERTAINMENTS </option>
</select>
```

<select>	Opening tag to create a drop-down list.
<option>	Defines available options in the list.
</option>	Closing option tag.
</select>	Select closing tag.

7.3.7 Search Button

Create a search button that will be used to search user-entered text.It will bea simple button with title. Later, you can add action to this button so search can be performed on the entered text from the search bar.

```
<button>Search</button>
```

<button>	Defines a clickable button.
</button>	Button closing tag.

7.3.8 Shopping Cart

Create a dedicated box to show a shopping cart. This cart will contain total number of items that user has added while shopping on the site. In addition, this box will also show the total price for all the products that are in the cart.

```
<span ><i> [ 0 ] Items in your cart </i></span>
```

	Useful to group in-line elements in a document.
<i>	Displays a text in italic.
</i>	Italic closing tag.
	Span closing tag.

7.3.9 Navigation Menu – Home

Create a home button for the website.

```
<a href="home.html">Home</a>
```

<a>	Defines a clickable anchor link.
href	This attribute sets the resource URL of page which will load upon clicking on this anchor tag.
**	Anchor link closing tag

7.3.9.1 Navigation Menu – Special Offers

Create a button dedicated to view special offers and discounts page

```
<a href="specialoffers.html">Special Offers</a>
```

7.3.9.2 Navigation Menu – Contact

Create a button dedicated to view contact information

```
<a href="contact.html">Contact</a>
```

7.3.9.3 Navigation Menu – Login

Create a button dedicated to open Login and sign up page.

```
<a href="login.html">Login</a>
```

7.3.10 Execution of Code

Now let us execute the final HTML code:

```
<!DOCTYPEhtml>
<html>
    <head>
        <title>MyEShop</title>
    </head>
    <body>
        <h1>Welcome to MyEShop</h1>
        <p>Buy Any Product you Like at a Reasonable Price</p>

        <table>
            <tr>
                <td><imgsrc="myeshoplogo.png"alt="myeshop logo"width="100px"
height="50px"></td>
                <td><form><inputtype="text" placeholder="Search.."></form></td>
                <td>
                    <select>
                        <option>All</option>
                        <option>CLOTHES </option>
                        <option>FOOD AND BEVERAGES </option>
                        <option>HEALTH & BEAUTY </option>
                        <option>SPORTS & LEISURE </option>
                        <option>BOOKS & ENTERTAINMENTS </option>
                    </select>
                </td>
                <td><button>Search</button></td>
                <td><span><i> [ 0 ] Items in your cart </i></span></td>
                <td><ahref="home.html">Home</a></td>
                <td><ahref="specialofferrs.html">Special Offers</a></td>
                <td><ahref="contact.html">Contact</a></td>
                <td><ahref="login.html">Login</a></td>
            </tr>
        </table>

    </body>
</html>
```

If you open this file in a browser, you will see the following result. Please note, we are not seeing any image in the place where we put logo, instead we see a text "*myeshop logo*". If you look closely, you will notice that "*myeshop logo*" is the alt text we use on the *<img>* tag. So, when it does not find the specified image, it shows the alt text instead.

Welcome to MyEShop

Buy Any Product you Like at a Reasonable Price

myeshop logo Search..   All   ⬍ Search   *[0] Items in your cart* Home Special Offers Contact Login

Based on the above example, continue the development according to the page flow diagram shown in Figure 7.2. Add more HTML elements to this page and create more pages according to the Navigation Bar. Take help of Entity Relationship Diagram to identify these pages and how to structure them. You may also create page flow diagram before starting the coding activity.

QUICK CHALLENGE Tom has created his first HTML page and added all the elements we have discussed so far in this chapter. Now, he would like to add Marquee Tag, Checkbox, and compare button. Help him code these elements.

7.4 | Adding CSS to the HTML Page

Now we will design the page and make it more presentable by adding CSS elements to it (we have already learned this in Chapter 4). First, we will add background color to the page so our page looks a little visually appealing.

7.4.1 Background Color

To add a background color to the entire page, we need to add it to the root element *<body>* of the page. For this type of inline style, we need to use "style" attribute on *<body>* element and inside "*style*" attribute set the background color.

```
<body style="background-color:#A2E9FF;">
```

See the following screenshot that shows the changed background color. Also note that *#A2E9FF* is a Hex Color Code. You can use various online websites to find a color code that you like.

7.4.2 Heading: Center Align

We can also use CSS to align text. See the following example which shows how to align text using inline CSS.

```
<h1 style="text-align:center;">Welcome to MyEShop</h1>
```

This text styling can be applied on any element. See the below example of *<p>* tag.

```
<p style="text-align:center;">Buy Any Product you Like at a Reasonable Price</p>
```

Above changes will bring the following changes to the pages. Please note that we have use "*text-align:center;*" to make the text look center aligned. Similarly, you can make text left, right, or justify. For that you can use "*text-align:left;*", "*text-align:right;*", "*text-align:justify;*".

7.4.3 Navigation Menu Color

Now, let us change navigation bar menu color and make it look little different. In this example, we will add more than one styling to the element. See the following code that will help us to make the link look like a button.

```
<a style="background-color: red;color: white;padding: 1em 1.5em;text-decoration: none;-
text-transform: uppercase;" href="home.html">Home</a>
```

The above code produces the following result. See the change in look compared to the other links on which we did not apply this styling.

Let us understand the style elements that we used.

1. **background-color:**This element sets the background color.
2. **color:** This element sets color for text.
3. **padding:** This element adds space on all four directions like left, right, top, and bottom.
4. **text-decoration:** This element dictates how anchor tag should be decorated.
5. **text-transform:** This element changes case for the text.

Now you can see how easy it is to change look and feel of a page with a very simple code.

You might have noticed the main title and subtitle are aligned center however the entire row which contains the logo, search bar, category dropdown, search button and navigation menu looks like left aligned. You can fix this by wrapping the <table> element in a div and setting its position to center like the following;

```
<div align="center"><table>….</table></div>
```

Repeat this process for all other anchor links and if needed change the color of your choice.

7.4.4 Fix the Image Problem

You might have noticed another problem like our <table> alignment and that is the <img> problem. Our logo is not shown, instead it is showing the alt attribute we have used on the image element. It is very easy to fix the image problem we are experiencingfor logo. As we have discussed earlier, this problem is there because browser does not find the mentioned image in the same directory as *home.html* file. Hence, to fix this problem, add the logo image named "*myeshoplogo.png*" to the same directory where "*home.html*" is placed.

See the following image which shows the image is clearly visible and alt text is hidden because browser can now find the image to display.

Finally, the code looks as follows:

```
<!DOCTYPEhtml>
<html>
    <head>
        <title>MyeShop</title>
    </head>
    <bodystyle="background-color:#A2E9FF;">
        <h1style="text-align:center;">Welcome to MyeShop</h1>
        <pstyle="text-align:center;">Buy Any Product you Like at a Reasonable Price</p>

        <table>
            <tr>
                <td><imgsrc="myeshoplogo.png"alt="myeshop logo"width="100px"
height="50px"></td>
                <td><form><inputtype="text" placeholder="Search.."></form></td>
                <td>
                    <select>
                        <option>All</option>
                        <option>CLOTHES </option>
                        <option>FOOD AND BEVERAGES </option>
                        <option>HEALTH & BEAUTY </option>
                        <option>SPORTS & LEISURE </option>
                        <option>BOOKS & ENTERTAINMENTS </option>
                    </select>
                </td>
                <td><button>Search</button></td>
                <td><span><i> [ 0 ] Items in your cart </i></span></td>
                <td><astyle="background-color: red;color: white;padding: 1em 1.5em;-
text-decoration: none;text-transform: uppercase;" href="home.html">Home</a></td>
                <td><ahref="specialofferrs.html">Special Offers</a></td>
                <td><ahref="contact.html">Contact</a></td>
                <td><ahref="login.html">Login</a></td>
            </tr>
        </table>

    </body>
</html>
```

Summary

We have learned how to read the architectural diagrams we have created in Chapter 2. The aim of this chapter is not to give you a finished web application but to give you a starting point to build your own.

In this chapter, we have learned the following concepts:

1. Reading architectural diagrams.
2. Understand Page Flow diagram and think of creating pages.
3. Start with boiler plate code and add various elements to get desired results.
4. Use of common HTML elements.
5. Add CSS to HTML elements to make them look good.

In Chapter 8, we will take one step further and add jQuery to this page to add some programming logic.

Multiple Choices Questions

1. Which of the following tags is used to indicate the beginning of a paragraph?
   - **(a)** <TR>
   - **(b)** <p>
   - **(c)**
   - **(d)** <TD>
2. Markup tags are used to inform the web browser about which of the following?
   - **(a)** How to display the page.
   - **(b)** How to display a message box on the page.
   - **(c)** How to organize the page.
   - **(d)** None of the above
3. The file of map definition is usually stored in which of the following?
   - **(a)** RECYCLE-bin
   - **(b)** BIN
   - **(c)** CGI-BIN
   - **(d)** All of the above
4. Which of the following CSS properties defines the bottom-left corner shape of the border?
   - **(a)** border-corner-radius
   - **(b)** border-radius
   - **(c)** border-left-radius
   - **(d)** border-bottom-left-radius
5. Which of the following CSS properties can be used for collapsing the borders between table cells?
   - **(a)** collapse-border
   - **(b)** border-collapse
   - **(c)** border-cell
   - **(d)** border

Review Questions

1. How do we use ERD to decide on creating HTML pages?
2. How can you add CSS to the HTML code?
3. How do you fix the image problem where the text given in the alt attribute is shown?
4. How do you align table element to center?
5. How do you create a multiple selection dropdown?

Exercise

Continue developing this further and add more and more styling to this page to make it attractive. Also add more pages and styling to those. In the end of the exercise, you will have a nice-looking ecommerce front design.

Project Idea

Take the example of a school management system in which school admins can add, update, and delete students. Teachers can view students' data and also grade students on class tests. Design front end pages in HTML and style them with CSS for the school management system.

Recommended Readings

1. Jon Duckett. 2011. *HTML and CSS: Design and Build Websites*. Wiley: New Delhi
2. Laura Lemay, Rafe Colburn, Jennifer Kyrnin .2016. *Mastering HTML, CSS & Javascript Web Publishing*. BPB Publications
3. Paul McFedries. 2018. *Web Coding & Development All-in-One For Dummies*. Wiley: New Delhi